

1 the defining of the plurality of states comprises defining one or more
2 schema modules that are configured to track one or more states of the XML data
3 stream; and

4 the evaluating comprises using the one or more schema modules to evaluate
5 the XML data stream against one or more schema-based rules.

6
7 4. The method of claim 1, wherein the defining of the plurality of states
8 comprises defining one or more schema modules that are configured to track one
9 or more states of the XML data stream, each schema module being associated with
10 at least one request type that defines the XML data stream.

11
12 5. The method of claim 4, wherein the request type is a WebDAV
13 request type.

14
15 6. The method of claim 5, wherein the WebDAV request type is a
16 PROPFIND request.

17
18 7. The method of claim 5, wherein the WebDAV request type is a
19 PROPPATCH request.

20
21 8. The method of claim 5, wherein the WebDAV request type is a
22 SEARCH request.

23
24 9. The method of claim 5, wherein the WebDAV request type is one of
25 a LOCK and UNLOCK request.

1 10. The method of claim 1 further comprising defining one or more rules
2 that relate to an element's contents.

3
4 11. The method of claim 10, wherein said one or more rules that relate
5 to an element's contents define which elements can be contained within other
6 elements.

7
8 12. The method of claim 11, wherein if a rule that defines which
9 elements can be contained within other elements is violated, disregarding
10 associated portions of the XML data stream until a close tag is received for an
11 element that violates the rule.

12
13 13. A computer-readable medium having a program thereon which,
14 when executed by a computer, performs the steps of claim 1.

15
16 14. A method of parsing an Extensible Markup Language (XML) data
17 stream comprising:

18 defining a schema module that is associated with an HTTP request type that
19 is received from a client, the schema module having a function that determines
20 whether an XML data stream conforms to a given schema that is associated with
21 the HTTP request type;

22 evaluating an XML data stream with the schema module; and

23 disregarding a portion of the XML data stream if it does not conform to the
24 given schema.
25

1 15. The method of claim 14, wherein said defining of the schema
2 module comprises defining a plurality of schema modules, individual schema
3 modules being associated with different HTTP request types.

4
5 16. The method of claim 14, wherein said function determines whether
6 there are any unauthorized elements that appear in a client's request.

7
8 17 The method of claim 14, wherein said function determines whether
9 there are any unauthorized elements that appear in a client's request; said
10 disregarding comprising disregarding said XML data stream portion until a close
11 tag is received for an unauthorized element.

12
13 18. The method of claim 14, wherein said HTTP request type comprises
14 a WebDAV request type.

15
16 19. The method of claim 18, wherein said WebDAV request type
17 comprises a PROPFIND request.

18
19 20. The method of claim 18, wherein said WebDAV request type
20 comprises a PROPPATCH request.

21
22 21. The method of claim 18, wherein said WebDAV request type
23 comprises a SEARCH request.

24
25 22. The method of claim 18, wherein said WebDAV request type
comprises one of a LOCK and UNLOCK request.

1
2 23. A computer-readable medium having a program thereon which,
3 when executed by a computer, performs the steps of claim 14.

4
5 24. An Extensible Markup Language (XML) parsing system comprising:
6 a parser configured to receive an XML data stream and generate a series of
7 calls as it parses the XML data stream;

8 a node factory communicatively associated with the parser and configured
9 to receive the parser's calls and responsive thereto construct a representation of the
10 XML data stream that the parser is parsing; and

11 a schema module communicatively associated with the node factory and
12 configured to evaluate the node factory's representation of the XML data stream
13 and determine whether it conforms to a known schema.

14
15 25. The parsing system of claim 24, wherein said parsing system
16 comprises a plurality of schema modules, each schema module being associated
17 with a different known schema.

18
19 26. The parsing system of claim 24, wherein the schema module
20 corresponds to an HTTP request type.

21
22 27. The parsing system of claim 24, wherein said parsing system
23 comprises a plurality of schema modules, each schema module being associated
24 with a different known schema and corresponding to a different HTTP request
25 type.

1 28. The parsing system of claim 27, wherein at least one of the different
2 HTTP request types is a WebDAV request.

3
4 29. The parsing system of claim 24, wherein the schema module is
5 configured to ignore an XML data stream portion that does not conform to the
6 known schema.

7
8 30. An Extensible Markup Language (XML) parsing system comprising:
9 a collection of schema modules, each of which being configured to evaluate
10 a different schema that is associated with an XML data stream; and
11 a plurality of states associated with each schema module, individual states
12 of a schema module defining a schema requirement relating to a particular element
13 that is evaluated by that schema module.

14
15 31. The parsing system of claim 30, wherein each schema module is
16 associated with a different HTTP request and is configured to evaluate a schema
17 that is associated with the HTTP request with which it is associated.

18
19 32. The parsing system of claim 31, wherein at least one of the HTTP
20 requests is a WebDAV request.

21
22 33. The parsing system of claim 31, wherein each of the HTTP requests
23 is a WebDAV request.